

CLAIMS

1. A computer-implemented simulation method comprising:

- modelling a target system as a set of processes that communicate with each other by way of messages;
- associating the messages with sender and receiver processes;
- scheduling the messages for processing; and
- processing each scheduled message by calling sender and receiver processes associated with the message.

2. A method according to Claim 1 wherein the step of associating the messages with sender and receiver processes comprises:

- providing a message data structure for each of the messages;
- when a process requires to send a message, adding that process to the relevant message data structure as a sender;
- when a process requires to receive a message, adding that process to the relevant message data structure as a receiver;

3. A method according to Claim 2, further including:

- when a process is added as a sender for a message, scheduling the message for processing only if at least one receiver for the message already exists in the message data structure; and
- when a process is added as a receiver for a message, scheduling the message for processing only if at least one sender for the message already exists in the message data structure..

4. A method according to Claim 2 wherein each message data structure includes a sender queue and a receiver queue, for queuing a number of sender and receiver processes.

5. A method according to Claim 2 further including checking the message data structure when a message has been processed, to determine whether there is at least one remaining sender and receiver for the message and, if so, rescheduling the message.

6. A method according to Claim 2 wherein each message data structure holds pointers to a composition activity, for composing a higher-level message from a lower-level message, and to a decomposition activity, for decomposing a higher-level message into a lower-level message.

7. A method according to Claim 6 wherein the composition activity performs multiple receives of the lower-level message followed by a single send of the higher-level message.

8. A method according to Claim 6 wherein the decomposition activity performs multiple sends of the lower-level message followed by a single receive of the higher-level message.

9. A method according to Claim 6 wherein:

- the decomposition activity for a message is activated when a process is added as a sender for that message; and
- the composition activity for a message is activated when a process is added as a receiver for that message.

10. A method according to Claim 1 wherein the step of scheduling the messages for processing comprises providing at least one scheduler queue, which is used for scheduling both messages and processes.

11. A method according to Claim 10 including the steps:

- scheduling the processes and messages by placing process-type items and message-type items on the scheduler queue;
- processing each process-type item on the scheduler queue by calling the process to which the item relates; and
- processing each message-type item on the scheduler queue by calling both the sender and receiver processes of the message to which the item relates.

12. A method according to Claim 10 wherein processes also communicate with each other by way of signals, and wherein said scheduler queue is used for scheduling the signals as well as the messages and processes.

13. A method according to Claim 12 including the steps:

- scheduling the signals by placing signal-type items on the scheduler queue; and
- processing each signal-type item on the scheduler queue by updating the signal state and calling all processes that are sensitive to the signal.

14. A method according to Claim 10 wherein there are two scheduler queues: an event queue for scheduling changes that are to take place at specified times, and a delta queue for scheduling changes that are to take place instantaneously.

15. A method according to Claim 1, including:

- providing an activity data structure for each of a plurality of activities, each activity data structure including a pointer to an associated activity process and a caller queue for queuing a number of caller processes;
- when a caller process requires to call an activity, adding the caller process to the caller queue of the related activity data structure, and scheduling the activity for processing;
- processing each scheduled activity by calling its associated activity process.

16. A method according to Claim 15 wherein the activity is scheduled only if it is not already scheduled for processing.

17. A method according to Claim 15 wherein, when an activity process finishes its processing:

- the corresponding caller process is removed from the caller queue; and
- if there are still any remaining caller processes in the caller queue, the activity is rescheduled.

18. A computer-implemented simulation method comprising the steps:

- modelling a target system as a set of processes that communicate with each other by way of messages;
- associating the messages with sender and receiver processes;
- providing at least one scheduler queue, holding a series of items, each item having a type value which indicates the item type;
- scheduling the processes and messages by placing process-type items and message-type items on the scheduler queue;
- processing each process-type item on the scheduler queue by calling the process to which the item relates; and
- processing each message-type item on the scheduler queue by calling both the sender and receiver processes of the message to which the item relates.

19. A data carrier, carrying a computer-readable program for performing a computer-implemented simulation method comprising:

- modelling a target system as a set of processes that communicate with each other by way of messages;
- associating the messages with sender and receiver processes;
- scheduling the messages for processing; and
- processing each scheduled message by calling sender and receiver processes associated with the message.